

trapfuzzer : coverage-guided binary fuzzing with breakpoint

Sili Luol @hac425

TRACK 2



About Me

- Security Researcher at Huawei ROOT Lab
- Focus on software vulnerability research
- Github: https://github.com/hac425xxx
- Twitter: https://twitter.com/hac425/
- Blog: https://www.cnblogs.com/hac425



Outline

1.Background

- 2.Implementation of trapfuzzer
- 3.How to Fuzz with trapfuzzer and results

4.Future Plans



Background





What is Fuzzing?

•••

```
def fuzzer(max_length=100, char_start=32, char_range=32):
    """A string of up to `max_length` characters
        in the range [`char_start`, `char_start` + `char_range`]"""
    string_length = random.randrange(0, max_length + 1)
    out = ""
    for i in range(0, string_length):
        out += chr(random.randrange(char_start, char_start + char_range))
    return out
```



What is Coverage-Guided Fuzzing?



Fig. 1: fuzz testing process.



Background

and achieves lightweight, low-overhead coverage guided fuzzing for closed source code by:

- 1. Enumerating the start offset of every basic block in the program/library. This is done with a IDAPython script
- 2. At runtime, in the fuzzed process, replacing the first byte of every undiscovered basic block breakpoint instruction (int3 on Intel). The original byte and the corresponding offset in the cibitmap are stored in a dedicated shadow memory mapping whose address can be compute address of the modified library, and
- 3. Installing a SIGTRAP handler that will:
 - a. Retrieve the faulting address and compute the offset in the library as well as the addr corresponding entry in the shadow memory
 - b. Mark the basic block as found in the global coverage bitmap
 - c. Replace the breakpoint with the original byte
 - d. Resume execution

CVE-2020-11764

An out-of-bounds write (of presumably image pixels) on the heap in the copyIntoFrameBuffer function.

CVE-2020-11763

A bug that caused a std::vector to be read out-ouf-bounds. Afterwards, the calling code would write into an element slot of this vector, thus likely corrupting memory.

CVE-2020-11762

An out-of-bounds memcpy that was reading data out-of-bounds and afterwards potentially writing it out-ofbounds as well.

CVE-2020-11760, CVE-2020-11761, CVE-2020-11758 Various out-of-bounds reads of pixel data and other data structures.

CVE-2020-11765

An out-of-bounds read on the stack, likely due to an off-by-one error previously overwriting a string null terminator on the stack.

CVE-2020-11759

https://googleprojectzero.blogspot.com/2020 /04/fuzzing-imageio.html

- Prep
 - IDA analysis
 - Basic blocks file generation
 - Modification of the exe/dll files
- Execution
 - Catch 0xCC exceptions
 - If in the basic block list
 - Record location
 - Replace 0xCC with original value
 - EIP = EIP 1

Vendor	CVE count
Microsoft	27
Adobe	45
Apple*	2

http://www.powerofcommunity.net/poc20 18/jaanus.pdf



Inspiration

Combining my previous fuzzing experience and these two security research, I realized:

- 1. The existing excellent Fuzz tools (AFL, honggfuzz) are not perfect, and there are still unsupported or incompletely supported scenarios, such as complex large, closed-source programs and some new platforms.
- 2. Using **relatively inefficient instrument methods** for fuzz testing can also obtain **better results** than complete black box fuzz testing, such as instruments with breakpoints.

I found that **fuzzing tools for large closed-source Linux software are rare**, commonly used linux fuzzing tools are Peach and AFL, they have some shortcomings:

- 1. Peach: fuzzing without coverage information.
- 2. AFL Qemu Mode: only suitable for relatively small programs, such as image parse library.

So I decided to develop a fuzzer based on a breakpoint mechanism to support some scenarios that are not covered by existing tools, such as large, closed-source file parsing programs and provide coverage support.



Implementation (Version 0.1)



HITB SECCONF. SIN-2021

Overview





binary patcher

- 1. Use IDAPython script to get all basic blocks of binary
- 2. Replace the first instruction of every basic blocks with breakpoint instruction and save the original instruction to **basic-block-info-file**.





binary patcher - basic-block-info-file example

JSTARO24.OCX-bb.txt ×

	0	1	2	3	4	5	6	7	Ś	9	A	В	C	D	E	F	0123456789ABCDEF
0000h:	00	60	B7	00	0D	00	00	00	4A	53	54	41	52	4F	32	34	JSTAR024
0010h:	2E	4F	43	58	00	00	10	00	00	00	04	00	00	01	00	00	.0CX.
0020h:	00	FF	06	10	00	00	06	04	00	00	01	00	00	00	FF	0C	.ÿÿ.
0030h:	10	00	00	0C	04	00	00	01	00	00	00	FF	12	10	00	00	ÿ
0040h:	12	04	00	00)	01	00	00	00	FF	18	10	00	00	18	04	00)ÿ
0050h:	00	01	00	00	00	FF	1E	10	00	00	1E	04	00	00	01	00	ÿ
0060h:	00	00	FF	24	10	00	00	24	04	00	00	01	00	00	00	FF	ÿ\$\$ÿ
0070h:	2A	10	00	00	2A	04	00	00	01	00	00	00	FF	30	10	00	**ÿ0
0080h:	00	30	04	00	00	01	00	00	00	FF	36	10	00	00	36	04	.0ÿ66.
0090h:	00	00	01	00	00	00	FF	3C	10	00	00	3C	04	00	00	01	ÿ<<
00A0h:	00	00	00	B8	44	10	00	00	44	04	00	00	01	00	00	00	,DD
00B0h:	B 8	A0	72	8E	00	A0	66	8E	00	01	00	00	00	8D	A8	72	, rŽ. fŽr
00C0h:	8E	00	A8	66	8E	00	01	00	00	00	B 8	68	1A	09	00	68	Ž. "fŽ,hh
00D0h:	0E	09	00	01	00	00	00	FF	CB	07	69	00	CB	FB	68	00	ÿË.i.Ëûh.
00E0h:	01	00	00	00	FF	8A	10	00	00	8A	04	00	00	01	00	00	ÿŠŠ
00F0h:	00	FF	90	10	00	00	90	04	00	00	01	00	00	00	FF	96	.ÿÿ-
0100h:	10	00	00	96	04	00	00	01	00	00	00	FF	90	10	00	00	ÿœ
0110h:	9C	04	00	00	01	00	00	00	FF	A2	10	00	00	A2	04	00	œÿ¢¢
0120h:	00	01	00	00	00	FF	A8	10	00	00	A8	04	00	00	01	00	ÿ¨¨
0130h:	00	00	8B	AF	10	00	00	AF	04	00	00	01	00	00	00	8B	
0140h:	C0	10	00	00	C0	04	00	00	01	00	00	00	8B	DO	10	00	ÀÀ

Template Results - bb.bt @

Name	Value	Start	Size	Co	lor
uint32 rva_size	B76000h	Oh	4h	Fg:	Bg:
✓ struct MODULE_NAME module_name	JSTARO24.OCX	4h	11h	Fg:	Bg:
uint32 length	13	4h	4h	Fg:	Bg:
> ubyte data[13]		8h	Dh	Fg:	Bg:
✓ struct BB_INFO bi[0]	id:0, rva:0x1000, foff:0x400, instr size:0x1	15h	Dh	Fg:	Bg:
uint32 rva	4096	15h	4h	Fg:	Bg:
uint32 foff	1024	19h	4h	Fg:	Bg:
uint32 instr_size	1	1Dh	4h	Fg:	Bg:
> ubyte instr[1]		21h	1h	Fg:	Bg:
> struct BB_INFO bi[1]	id:1, rva:0x1006, foff:0x406, instr size:0x1	22h	Dh	Fg:	Bg:
> struct BB_INFO bi[2]	id:2, rva:0x100c, foff:0x40c, instr size:0x1	2Fh	Dh	Fg:	Bg:
> struct BB_INFO bi[3]	id:3, rva:0x1012, foff:0x412, instr size:0x1	3Ch	Dh	Fg:	Bg:
> struct BB_INFO bi[4]	id:4, rva:0x1018, foff:0x418, instr size:0x1	49h	Dh	Fg:	Bg:
> struct BB_INFO bi[5]	id:5, rva:0x101e, foff:0x41e, instr size:0x1	56h	Dh	Fg:	Bg:
> struct BB_INFO bi[6]	id:6, rva:0x1024, foff:0x424, instr size:0x1	63h	Dh	Fg:	Bg:
> struct BB INFO bi[7]	id:7, rva:0x102a, foff:0x42a, instr size:0x1	70h	Dh	Fg:	Bg:



binary patcher - example

			• •		• •			_				
俞主	页					× 📳 et_original <> et	×					
☆ 主页		米 全部	≠ 差别	目同規		協議 ↓ ↓ 格式 ↓ ↓ 复制 下一个对比						
	E:\vuln_o	data\wps	_fuzzing	\10161\et	\et_or	iginal		> → →	E:\vuln_data\wps_fuzz	zing\10161\et\et		
	2021/1,	/15 16:21	1:01 1,8	76,128字	节其	挖一切 ▼			2021/1/15 16:21:01	1,876,128 字节 其	蛇一切 ▼	
	000D534	C E2 ØF	39 D6	73 2C 4	5 ØF	B6 61 01 89 F1 49 D3 E4	â.9Ös, <mark>E</mark> .¶a.‰ñIÓä	^	000D534C E2 0F 39	D6 73 2C CC 0F	B6 61 01 89 F1 49 D3 E4	â.9Ös, <mark>Ì</mark> .¶a.‰ñIÓä
	000D535	C 49 01	DC 8D	46 08 3	9 D0	73 1F 41 0F B6 59 02 89	I.Ü.F.9Ðs.A.¶Y.‰		000D535C 49 01 DC	8D 46 08 39 D0	73 1F CC 0F B6 59 02 89	I.Ü.F.9Ðs.Ì.¶Y.‰
	000D536	C C1 48	B D3 E3	49 83 C	1 02	49 01 DC 83 C6 10 89 F0	ÁHÓãIƒÁ.I.܃Æ.‰ð		000D536C C1 48 D3	E3 49 83 C1 02	49 01 DC 83 C6 10 89 F0	ÁHÓãIƒÁ.I.܃Æ.‱ð
	000D537	C EB ØE	8 <mark>89</mark> FØ	49 89 D	C EB	04 49 83 <mark>C1</mark> 01 BB FF FF	ë. <mark>‰</mark> ðI‰Üë. <mark>I∱Á</mark> .»ÿÿ		000D537C EB 0B CC	F0 49 89 DC EB	04 CC 83 C1 01 CC FF FF	ë. <mark>Ì</mark> ðI‰Üë.Ì∫Á.Ìÿÿ
	000D538	C FF FF	89 D1	D3 E3 F	7 D3	44 21 E3 44 8D 14 2B 49	ÿÿ‱ÑÓã÷ÓD!ãD+I		000D538C FF FF 89	D1 D3 E3 F7 D3	44 21 E3 44 8D 14 2B 49	ÿÿ‱ÑÓã÷ÓD!ãD+I
	000D539	C D3 EC	29 D0	44 89 D	9 2B	4C 24 50 44 89 D6 41 29	Óì)ĐD‰Ù+L\$PD‰ÖA)		000D539C D3 EC 29	DØ 44 89 D9 2B	4C 24 50 44 89 D6 41 29	Óì)ĐD‰Ù+L\$PD‰ÖA)
	000D53A	C CA ØF	86 4D	01 00 0	0 44	3B 54 24 F4 76 12 48 8B	Ê.†M <mark>D</mark> ;T\$ôv. <mark>H</mark> <		000D53AC CA 0F 86	4D 01 00 00 <mark>CC</mark>	3B 54 24 F4 76 12 CC 8B	Ê.†M <mark>Ì</mark> ;T\$ôv. <mark>Ì</mark> <
	000D53E	C 4C 24	1 EØ 83	B9 E0 1	B 00	00 00 0F 85 84 0A 00 00	L\$àf¹à,,		000D53BC 4C 24 E0	83 B9 E0 1B 00	00 00 0F 85 84 0A 00 00	L\$àf¹à,,
	000D530	C 48 8E	3 14 24	85 D2 4	8 89	74 24 C8 0F 84 B5 01 00	<mark>Η</mark> <.\$…ÒH‰t\$È.,,μ		000D53CC CC 8B 14	24 85 D2 48 89	74 24 C8 0F 84 B5 01 00	Ì<.\$…ÒH‰t\$È.,,μ
	000D53D	C 00 89	D1 44	29 D1 0	F 83	55 02 00 00 <mark>8B</mark> 4C 24 F0	.‱ÑD)Ñ.ƒU <l\$ð< th=""><th></th><th>000D53DC 00 CC D1</th><th>44 29 D1 0F 83</th><th>55 02 00 00 <mark>CC</mark> 4C 24 F0</th><th>.<mark>Ì</mark>ÑD)Ñ.fU<mark>Ì</mark>L\$ð</th></l\$ð<>		000D53DC 00 CC D1	44 29 D1 0F 83	55 02 00 00 <mark>CC</mark> 4C 24 F0	. <mark>Ì</mark> ÑD)Ñ.fU <mark>Ì</mark> L\$ð
	000D53E	C 44 29	D1 48	89 D6 4	8 8B	54 24 98 48 01 CA 41 29	D)ÑH‱ÖH <t\$~h.êa)< th=""><th></th><th>000D53EC 4 29 D1</th><th>48 89 D6 48 8B</th><th>54 24 98 48 01 CA 41 29</th><th>D)ÑH‰ÖH<t\$<sup>~H.ÊA)</t\$<sup></th></t\$~h.êa)<>		000D53EC 4 29 D1	48 89 D6 48 8B	54 24 98 48 01 CA 41 29	D)ÑH‰ÖH <t\$<sup>~H.ÊA)</t\$<sup>
	000D53F	C F2 44	1 89 FE	44 29 D	6 ØF	86 DB 02 00 00 89 74 24	òD‰þD)Ö.†Û <mark>‰</mark> t\$		000D53FC F2 44 89	FE 44 29 D6 0F	86 DB 02 00 00 CC 74 24	òD‱þD)Ö.†Û <mark>Ì</mark> t\$
	000D540	C 90 03	3 5C 24	EC 01 E	B 41	89 DF 45 29 DF 49 83 C7	\\$ì.ëA‰ßE)ßIfÇ		000D340C 90 03 5C	24 EC 01 EB 41	89 DF 45 29 DF 49 83 C7	\\$ì.ëA‰ßE)ßIfÇ
	000D541	.C 01 49	83 FF	20 72 3	2 49	8D 73 01 44 29 DB 48 8D	.I∫ÿ r2 <mark>I</mark> .s.D)ÛH.		000D541C 01 49 83	FF 20 72 32 CC	8D 73 01 44 29 DB 48 8D	.Ifÿ r2 <mark>l</mark> .s.D)ÛH.
	000D542	C 2C 19	48 03	6C 24 D	8 48	39 EE 0F 83 45 03 00 00	,.H.1\$ØH9î.ƒE		000D542C 2C 19 48	03 6C 24 D8 48	39 EE 0F 83 45 03 00 00	,.H.1\$ØH9î.fE
	000D543	IC <mark>49</mark> 80) 34 1B	48 83 C	6 02	48 8B 5C 24 B8 48 01 CB	<mark>I</mark> .4.HƒÆ.H‹\\$,H.Ë		000D543C CC 8D 34	1B 48 83 C6 02	48 8B 5C 24 B8 48 01 CB	Ì.4.HƒÆ.H<\\$,H.Ë
	000D544	C 48 39	F3 0F	83 2C Ø	3 00	00 <mark>4C</mark> 8B 7C 24 80 <mark>41</mark> 8D	H9ó.f,L< \$€ <mark>A</mark> .		000D544C 48 39 F3	0F 83 2C 03 00	00 CC 8B 7C 24 80 CC 8D	H9ó.f, <mark>Ì</mark> < \$€ <mark>Ì</mark> .
	000D545	ic <mark>6A F</mark> F	41 F6	C2 07 74	4 27	44 89 D3 83 E3 07 31 F6	jÿAöÂ.t' <mark>D</mark> ‱Ó∱ã.1ö		000D545C 6A FF 41	F6 C2 07 74 27	CC 89 D3 83 E3 07 31 F6	jÿAöÂ.t' <mark>Ì</mark> ‰Ófã.1ö
	000D546	C OF 1F	40 00	0F B6 40	C 32	01 41 88 4C 33 01 48 83	@. <mark>.</mark> ¶L2.A^L3.Hƒ		000D546C 0F 1F 40	00 CC B6 4C 32	01 41 88 4C 33 01 48 83	@.̶L2.A^L3.Hƒ
	000D547	C C6 01	1 39 F3	75 EE 4	9 01	F3 41 29 F2 48 01 F2 83	Æ.9óuî <mark>I</mark> .óA)òH.ò <mark>f</mark>		000D547C C6 01 39	F3 75 EE CC 01	F3 41 29 F2 48 01 F2 <mark>CC</mark>	Æ.9óuî <mark>ľ</mark> .óA)òH.ò <mark>ľ</mark>

breakpoint instruction in x86



binary patcher - example

.text:0000000000400B36	mov	rbp, rsp			
.text:0000000000400B39	sub	rsp, 30h			
.text:0000000000400B3D	mov	[rbp+argc], edi			
.text:0000000000400B40	mov	[rbp+argv], rsi unpatticu			
.text:000000000000844	mov	rax, fs:28h			
.text:0000000000400B4D	mov	[rbp+var_8], rax			
.text:0000000000400B51	xor	eax, eax			
.text:0000000000000053	cmp	[rbp+argc], 1	.text:000000000400B36	mov	rbp. rsp
.text:00000000000000000000000000000000000	jg	short loc_400B7C	text:0000000000400B39	sub	rsp. 30h
.text:00000000000000859	mov	rax, [rbp+argv]	text:00000000000000000000000000000000000	mov	[rbp-24h], edi Datched
.text:0000000000400B5D	mov	rax, [rax]	text:00000000000000000000000000000000000	mov	[rbp-30h], rsi
.text:0000000000400B60	mov	rsi, rax	text:00000000000000000000000000000000000	mov	ray fs:28h
.text:000000000000863	mov	<pre>edi, offset aSyntaxSInputFi ; "Syntax: %s <input file.\n<="" pre=""/></pre>	text:00000000000000000000000000000000000	mov	[rbn-8] ray
.text:000000000000868	mov	eax, 0	text:00000000000000000000000000000000000	vor	
.text:0000000000400B6D	call	_printf	toxt:00000000000000000000000000000000000	cmp	dword str [shp_24b] 1
.text:000000000000872	mov	edi, 1 ; status	toxt:000000000400055	Cilip	aword per [Top-24i1], I
.text:0000000000000077	call	_exit	. LEXT. 000000000000000000000000000000000000	JB	Short Ioc_400B/C
.text:0000000000087C ;			. LEXT: 000000000000000000000000000000000000	Inc	; map to bebugger
.text:00000000000000000000000000000000000			. LEX1:000000000000005A	mov	eax, [rop-son]
.text:00000000000000000000000000000000000		; CODE XREF: main+22↑j	.text:00000000000000000000000000000000000	mov	rax, [rax]
.text:00000000000000000000000000000000000	mov	[rbp+len], 0	.text:00000000000000000000000000000000000	mov	rsi, rax
.text:00000000000083	mov	rax, [rbp+argv]	.text:0000000000000063	mov	edi, offset asyntaxSinputFi ; Syntax: %s <input< td=""></input<>
.text:00000000000887	add	rax, 8	.text:000000000400B68	mov	eax, 0
.text:0000000000088B	mov	rax, [rax]	.text:0000000000400B6D	call	sub_400770
.text:0000000000088E	lea	rdx, [rbp+len]	.text:00000000000000872	mov	edi, 1
.text:000000000000892	mov	rsi, rdx ; len	.text:0000000000400B77	call	sub_4007F0
.text:000000000000895	mov	rdi, rax ; path	.text:0000000000000070		
.text:000000000000898	call	read_to_buf	.text:00000000000000000000000000000000000	2000	; CODE XREF: main:22†j
.text:0000000000000000	mov	[rbp+data], rax	.text:00000000000087C	int	3 ; Trap to Debugger
.text:000000000000000	mov	rax, [rbp+data]	.text:000000000000000 main	endp	
.text:00000000000400BA5	movzx	eax, byte ptr [rax]	.text:00000000000000000000000000000000000		
.text:0000000000008A8	movsx	eax, al	.text:0000000000400B7D ;		
.text:0000000000400BAB	sub	eax, 41h	.text:0000000000000B7D	db	45h
.text:0000000000400BAE	cmp	eax, 4 ; switch 5 cases	.text:000000000000B7D	in	al, dx
.text:0000000000400BB1	ja	<pre>loc_400C7F ; jumptable 000000000400BC1 default cas</pre>	^{;e} .text:0000000000400B7D ;		
.text:0000000000400BB7	mov	eax, eax	.text:00000000000400B7F	db e	3
			.text:0000000000000880 gword 400B80	dq 48D6	0458B48000000h, 8D48008B4808C083h, 0C78948D68948EC



Seed Mutation



data mutate example

RadamsaMutator

use radamsa to generate mutated testcases https://gitlab.com/akihe/radamsa

TinyMutator

- 1.support basic mutation strategies, such as byte
 flipping and random insertion of boundary values
 (such as 0xFFFFFFFF).
- 2. supporting config the variation ratio of mutation.



Fuzzer Module

Workflow

- 1. First load the initial testcases to seed queue and do corpus distillation on seed queue.
- 2. Then it will traverse the seed queue, mutate the testcase, then use the trace module to start the target process, finally get the coverage and execution status of the process.
- 3. If new coverage is found, the new testcase will be added to the seed queue.
- 4. If crash is found, the crash information is saved, such as registers, stack trace.

Scheduling strategies

- The score of testcases that discover new coverage will increase ↑
- The score of testcases that trigger DOS will decrease



Fuzzer Module - Corpus Distillation

The workflow of corpus distillation is as follows:

1. let the program process the testcases in the seed queue one by one.

2. then only save the testcases that can generate new coverage.



original seed queue



Trace module - Theory

Prep

- Use bb-patcher module.
- Replace the first instruction of basic block with breakpoint instruction (0xcc)

Execution

- 1. Catch 0xCC exceptions
- 2. Record location
- 3. Replace 0xCC with original value
- 4. Let process continue





Trace module #1 (PythonPtraceTracer)

- 1. First use create_and_attach_process to create the target process
- 2. Use **cont** to let the process continue, and use **waitSignals** to wait for the process to trigger signals, such as SIGABORT, SIGTRAP.
- 3. If the process triggers the SIGTRAP signal, record the value of the PC at this time and replace the breakpoint instruction with the original instruction
- 4. then let the process continue to execute, goto 2





Lets Fuzz WPS

WPS is an office processing software in China that supports viewing and editing DOC, XLS, PPT and other files



Cross-platform Office Suite

- trapfuzzer collect coverage by patch breakpoint instruction to target module
- therefore, it is necessary to find which module need be patched, that is, the module that is responsible for parsing the file



Lets Fuzz WPS - Find Target Module

In Windows platform, we can use **process monitor** to monitor the behavior of the program at runtime, and be able to obtain the **call stack**, by using this information can quickly locate the **data processing module**.

21:54 🛽	wps.exe	1284	🛃 UnlockFileSi	C:\input.doc	SUCCESS	Offset	11 15	kernel32 dll	ReadFile + 0x54	0v75a93f07
21:54 5	wps.exe	1284	🛃 LockFile	C:\input.doc	SUCCESS	Exclus	11 10	.1.20 411	Status Changes 1 Outsh	076145760
21:54	vps.exe	1284	LockFile	C:\input.doc	SUCCESS	Exclus	0 16	01eJ2. dll	StgUpenStorage + UXIED	UX (6045/19
21:54	wps.exe	1284	LockFile	C:\input.doc	SUCCESS	Exclus	U 17	ole32.dll	StgOpenStorage + Oxf96	0x76d457a4
21:04	wps.exe	1284	M LOCKFIle	C:\input.doc	SUCCESS	Offeet	U 18	ole32.dll	WriteClassStm + Ox78ae	0x76dbb364
21:54	wps.exe	1284	JunlockFileSi.	C:\input.doc	SUCCESS	Offset	U 19	ole32.dll	WriteClassStm + 0x774f	0x76dbb205
21:54 🔻	wps.exe	1284	🛃 UnlockFileSi	C:\input.doc	SUCCESS	Offset	1 20	ole32 dll	WriteClassStm + Dv7638	Ov76dbb0ee
21:54 🔊	vps.exe	1284	🛃 UnlockFileSi	C:\input.doc	SUCCESS	Offset	1 01	01632.011		o repose
21:54	vps.exe	1284	🛃 ReadFile	C:\input.doc	SUCCESS	Offset	0 21	ole32.dll	WriteLlassStm + Ux4b9i	Ux /6db8655
21:54	wps.exe	1284	LockFile	C:\input.doc	SUCCESS	Exclus	U 22	ole32.dll	StringFromIID + 0x6e0	0x76d44476
21:54	wps.exe	1284	M LOCKFile	C:\input.doc	SUCCESS	Exclus	U 23	ole32.dll	StringFromIID + 0x882	0x76d44618
21:54	wps.exe	1204	Wuerystandar.		SULLESS	ALIOCI	24	ole32.dll	SteOpenStorage + 0x27	0x76144835
21:54.	wps.exe	1284	QueryBasicIn.		SUCCESS	Creat:	11 or	1 111		0.000.71
21:14 🦿	g wps.exe	1284	ReadFile	C:\input.doc	SUCCESS	Offset	0 25	KSO. dll	kso_iryopenincbackupFile + 0x25	Uxbac9e/le
21:54 🛛	wps.exe	1284	🛃 ReadFile	C:\input.doc	END OF FILE	Offset	U 26	wpsmain.dll	WpsTextboxFactory::createLayerControl + 0x7c235	0x64 <mark>1</mark> d595b
21:54	wps.exe	1284	🛃 UnlockFileSi	C:\input.doc	SUCCESS	Offset	U 27	wpsmain.dll	TypoLib::TypoView::setExportType + 0x100c9	0x64 <mark>1</mark> 42299
21:54	wps.exe	1204	CloseFile	C:\input.doc	SUCCESS	orrsei	U 28	wpsmain.dll	TypoLib::TypoView::setExportType + 0xe963	0x64 <mark>1</mark> 40b33
21:54	wps.exe	1284	🛃 CreateFile	C:\input.doc	SUCCESS	Desire	U 29	wpsmain.dll	TypoLib::TypoView::setExportType + Oxdbc1	0x64 <mark>1</mark> 3fd91
21:54 🔊	wps.exe	1284	🛃 QueryBasicIn	C:\input.doc	SUCCESS	Creat:	1 30	umenain dll	Typolib .: Typoliew .: set Export Type + Oydac7	0x64 3fc97
21:54 🔊	v wps.exe	1284	🛃 CloseFile	C:\input.doc	SUCCESS			*powdin dii		0. 64106714
21:54	vps.exe	1284	CreateFile	C:\input.doc	SUCCESS	Desire	0 31	wpsmain.dll	lypoLiD::lypoview::setExportlype + Uxd5e4	UX 641. JT 7 D4
21:54	wps.exe	1284	QueryBasicin.	C:\input.doc	SUCCESS	Creat:	U 32	wpsmain.dll	TypoLib::TypoView::setExportType + 0xd005	0x64 <mark>1</mark> 3f1d5
21:04	wps.exe	1284	CrosteFile	C:\input.doc	SUCCESS	Decire	U 33	wpsmain.dll	TypoLib::TypoView::setExportType + 0xc552	0x64 <mark>1</mark> 3e722
21:54	wns.exe	1284	QueryBasicIn.	C:\input.doc	SUCCESS	Creat	1 34	kso. dll	KxDrTinWidget::resizeEvent + 0x6f7	0x68c83d2c
21:54	wps.exe	1284	CloseFile	C:\input.doc	SUCCESS		11.25	hee dll	Kullain Kinden and free Kullan + 0x251	069.9602
21:54 🔊	wps.exe	1284	🛃 CreateFile	C:\input.doc	SUCCESS	Desire	0 30	kso.ull	KXMainwindow::notifyopenFiles + 0x201	0x00000002
21:54 🖻	vps.exe	1284	🛃 QueryBasicIn	C:\input.doc	SUCCESS	Creat:	U 36	kso.dll	KPromeServicesHandler::openFiles + 0x9c0	0x68c85b2f
21:54	vps.exe	1284	🛃 CloseFile	C:\input.doc	SUCCESS		U 37	kso.dll	GlobalFilterMgr::operator= + 0x4c7bc	0x69 <mark>5</mark> 50315
21:54	wps.exe	1284	CreateFile	C:\input.doc	SUCCESS	Desire	U 38	kso.dll	GlobalFilterMgr::operator= + 0x2f674	0x69 <mark>5</mark> 331cd
21:54	wps.exe wps.exe	1284	CreateFile	C:\input.doc	SUCCESS	Desire	U 39	kso.dll	KPromeServicesHandler::KPromeServicesHandler + 0x2f7f	0x691d0f3e
21:54	wps.exe	1284	🛃 QueryBasicIn.	C:\input.doc	SUCCESS	Creat:	11 40	has dil	CdimEmfToWmfBito + 0w2012	060560697
21:54 🔊	wps.exe	1284	🛃 CloseFile	C:\input.doc	SUCCESS		0 40	ksu.ull	GUIDEWLIDUUS + OXJOIJ	0x0900101
21:54 🛛	vps.exe	1284	🛃 CreateFile	C:\input.doc	SUCCESS	Desire	041	KSOIICE. UII	RTCClientChannelonRecvData + 0x300	0x121f345f
21:54 🔊	wps.exe	1284	🛃 QueryBasicIn	C:\input.doc	SUCCESS	Creat:	U 42	ksolite.dll	KLiteStyleOptionShadowLine:: ~KLiteStyleOptionShadowLine + 0x196	0x721d89e2
21:54	wps.exe	1284	CloseFile	C:\input.doc	SUCCESS	Destin	U 43	user32.dll	gapfnScSendMessage + 0x332	0x759962fa
21:04	wps.exe	1284	UnervBasicTr	C:\input.doc	SUCCESS	Creati	1 44	user32 d11	CetThreadDeckton + 0xd7	0.75006432
21:54	wps.exe	1284	CloseFile	C:\input.doc	SUCCESS	CICAL.	0 44	488132.411		0
21:54	wps.exe	1284	CreateFile	C:\input.doc	SUCCESS	Desire	U 45	user32.d11	CharPrevW + Ux138	Ux759977c4
Contraction and the		100 North 100			A STATE OF A		1 40	00 122	D:	0.5500500



Lets Fuzz WPS

What about the Linux platform?



- 1. GDB can obtain the call stack stably
- File operations of the process can be tracked through breakpoints
- 3. Use GDB's scripting mechanism to automate



Lets Fuzz WPS - Linux Version of FileMon

class FopenGoodFile(gdb.Breakpoint): def __init__(self, name):

super(FopenGoodFile, self).__init__(
 name, gdb.BP_BREAKPOINT, internal=False)
self.hitcount = 0

def stop(self):

```
rdi = read_register("rdi")
fname = read_memory(rdi, 0x100)
# print("open: {}".format(fname))
fname = fname[:fname.find(b"\x00")]
```

if b".doc" in fname:

```
current_frame = gdb.selected_frame()
caller = current_frame.older().pc()
print("[FopenGoodFile] open {}, set bp on
      fname, caller))
get_backtrace()
AddFpBp("*{}".format(caller))
return False
```

readGoodFp] fread fp: 62681600 0x00007ffff69bb3f0 in _____GI___IO_fread (buf=0x6cc2041, size=1, count=49147, fp=0x3bc7200) at iofread.c:31 0x00007fffec241e1c in XMLPlatformUtils::readFileBuffer(void*, unsigned int, unsigned char*) () at /opt/kingsoft/wps-office6/libkso 0x00007fffec249311 in XMLReader::refreshCharBuffer() () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec26c49d in ReaderMgr::peekNextChar() () at /opt/kingsoft/wps-office/office6/libkso.so @x00007fffec25da68 in XML5canner::scanProlog() () at opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec25d6b4 in XMLScanner::scanDocument(InputSource const&) () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec25d2fd in XMLScanner::scanDocument(unsigned short const*) () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec351743 in SAX2XMLReaderImpl::parse(unsigned_short_const*)_()_at_/opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec43d37e in () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec41e8a3 in OpenXmlPackage::_Open(unsigned short const*, unsigned int, OpcPackageIOType) () at /opt/kingsoft/wps-office/office6 0x00007fffec5d6a04 in WordProcessDocument::Open(unsigned short const*, WordProcessDocument::DocumentType) () at /opt/kingsoft/wps-office/ 11 0x00007fffe7dcc120 in () at /opt/kingsoft/wps-office/office6/libwpsio.so 12 0x00007fffe7d7aaf8 in filterpluginFormatCorrect_docx () at /opt/kingsoft/wps-office/office6/libwpsio.so 13 0x00007ffff357cc3b in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 14 0x00007ffff358815e in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 07ffff3Sa4cca in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 07ffff35Bd04b in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 007ffff35Bc502 in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 18 0x00007ffff35940bf in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 19 0x00007ffff3879ce1 in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 20 0x00007ffff305374b in () at /opt/kingsoft/wps-office/office6/libwpsmain.so ReadGoodFd] read fd: 29 oxooooriiiioa+ciao in __oi___libc_read (fd=29, buf=0x6cc2041, nbytes=45056) at ../sysdeps/unix/sysv/linux/read.c:27 0x00007ffff69c76e2 in __GI__IO_file_xsgetn (fp=0x3bc7200, data=<optimized out>, n=49147) at fileops.c:1364 0x00007ffff69bb431 in __GI__IO_fread (buf=<optimized out>, size=1, count=49147, fp=0x3bc7200) at iofread.c:38 0x00007fffec241eic in XMLPlatformUtils::readFileBuffer(void*, unsigned int, unsigned char*) () at /opt/kingsoft/wps-office/office6/libkso 0x00007fffec249311 in XMLReader::refreshCharBuffer() () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec26c49d in ReaderMgr::peekNextChar() () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec25da68 in XMLScanner::scanProlog() () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec25d6b4 in XMLScanner::scanDocument(InputSource const&) () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec25d2fd in XMLScanner::scanDocument(unsigned short const*) () at /opt/kingsoft/wps-office/office6/libkso.so 007fffec351743 in SAX2XMLReaderImpl::parse(unsigned short const*) () at /opt/kingsoft/wps-office/office6/libkso.so 0x00007fffec43d37e in () at /opt/kingsoft/wps-office/office6/libkso.so 07fffec41e8a3 in OpenXmlPackage::_Open(unsigned short const*, unsigned int, OpcPackageIOType) () at /opt/kingsoft/wps-office/office6

12 0x00007fffec5d6a04 in WordProcessDocument::Open(unsigned short const*, WordProcessDocument::DocumentType) () at /opt/kingsoft/wps-office/

13 0x00007fffe7dcc120 in () at /opt/kingsoft/wps-office/office6/libwpsio.so

IS 0x00007ffff357cc3b in () at /opt/kingsoft/wps-office/office6/libwpsmain.so

16 0x00007ffff358815e in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 17 0x00007ffff35a4cca in () at /opt/kingsoft/wps-office/office6/libwpsmain.so 18 0x00007ffff358044b in () at /opt/kingsoft/wps-office/office6/libwpsmain.so

14 0x00007fffe7d7aaf8 in filterpluginFormatCorrect_docx () at /opt/kingsoft/wps-office/office6/libwpsio.so



Lets Fuzz WPS

Related modules for processing doc files

```
"coverage_module_name": [
   "libkso.so",
   "libwpsapi.so",
   "libwpsmain.so",
   "wps"
],
```



Windows







Failure and plan

WPS can't execute within python-trace!

I decided to develop the trace module based on gdb.

The reasons are as follows:

1.Stable, few bugs, support multiple platforms and architectures

2.Support develop plugin with python

3.Open source, can be customized on demand



Implementation

(Version 0.2 - GdbPythonPluginTracer)





GDB Python API

Introduces some commonly used API of GDB Python Plugin

gdb.selected_frame().read_register(): read register value gdb.selected_inferior().write_memory(): write process memory gdb.selected_inferior().read_memory(): read process memory gdb.parse_and_eval(): execute gdb expression and get the result of expression gdb.execute(): execute gdb command and get the result of command execution

gdb.events.stop.connect: register gdb's stop event callback function, for example, when the process
triggers a signal, it will enter the callback function for processing.



Workflow

Due to the limitation of the gdb python script, the fuzzer needs to continuously use stdin and stdout to interact with gdb during the test.





Code - GDB Plugin

trap.py

```
def stop_handler(event):
    elif isinstance(event, gdb.StopEvent):
        pc = get_register("$pc") - 1
       hit_mod = None
        for mt in module_trace_list:
            if pc > mt['image_base'] and pc < mt['image_end']:</pre>
                hit_mod = mt
                break
       offset = pc - hit_mod['image_base']
       raw_byte = get_raw_byte_by_offset(offset)
        # restore original instructions, then set pc pc - 1
       write_memory(pc, raw_byte, 1)
       set_register("pc", pc)
       # log executed basic block
       mt['bbl-list'].append(offset)
    else:
        print("Unknown event {}".format(event))
# register callback
gdb.events.stop.connect(stop_handler)
```



Code - Tracer Part

gdb-python-plugintracer.py

```
def exec_with_gdb(self, timeout=30):
    command = "/usr/bin/gdb -q -x {}/cmd.gdb --args {}".format(self.workspace, self.cmdline)
    self.p = subprocess.Popen(command, shell=True, cwd=self.workspace, stdin=subprocess.PIPE,
                              stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
    # for dos
    timer = Timer(timeout, self.timeout_handler)
    try:
        timer.start()
       while True:
            l = self.p.stdout.readline()
           # process hit breakpoint
           if "received signal SIGTRAP" in 1:
                self.p.stdin.write("c\n")
            # test finished
            elif "[trapfuzzer] save_bb_trace" in 1:
                break
    except Exception as e:
        pass
    finally:
        timer.cancel()
    self.p.kill()
    self.p.wait()
```



When to kill target process

Problem & Common Solution

WPS will not exit the program after parsing the file, but will stay on the GUI interface and wait for the user to operate, so the **fuzzer needs to manually kill the process**. The commonly used method is to set a timeout, when the timeout occurs, the fuzzer will kill the

process.

Disadvantage

The execution time of the program is fixed each time. It will waste time when process simple testcase (actually parsing time is less than timeout), and when process complex testcase (actually parsing time is greater than timeout), it will result in an incomplete file parsing.
 Unable to detect DOS vulnerabilities.



When to kill target process - our solution

- First, use the trace module to trace the program execution and print the basic blocks executed by the program, and find the last basic block END_BBL executed after the process has parsed the input file.
- 2. In the following fuzzing, when the process reaches END_BBL, it means that the process has entered the GUI loop, and the process can be killed at this time.

During the fuzzing, record the average time (avg_time) of each testcase, and then set the DOS timeout to 10* avg_time.

When the execution time is greater than timeout, it is considered that DOS bug has been found.



When to kill target process

																	· U ^
5卡(T) 帮助)(H)	- 4	ρ 🚇 🖇			22	>_	7									
命 主页	× G	pwn_iot ×	win10-wps-fu	zzing >	kal	li-linux-2021	.1-vmware	e-amd64	×	🔓 kali-fi	uzzer X	🕞 ubuntu1804 🗙					
Activities	П.Т.	erminal x			L.F						т	ie 18:11 •				± 1	1 () -
Accivicies		erminat															* 0 *
	-																
		> 4 습 Ho	me et-fu	zzing 🕨							🍈 C						
			-														
	Ø	Recent		5)													
	企	Home											hac425	5@ubuntu: ~/t	rapfuzzer-gdb/build		
		Desktop	Inp	ut.xts								File Edit View Sea	arch Terminal Ta	abs Help			
		De contra										hac425@ubuntu: ~/tr	ranf x bac4	125@ubuntur -	/wps/ x bac425@ubur	tur /tranf X	
		Documents										[tranfuzzer] pate	ch to libkso.	so10x2773B4	.0	cu. yrrapr	
>_	÷	Downloads										[trapfuzzer] pate	ch to libkso.	so!0x2773B4	6		
	53	Music										[trapfuzzer] pate	ch to libkso.	so!0x2773B4	A		
	6	Pictures										[trapfuzzer] pate	ch to libkso.	so!0x2773CA	.6		
												[trapfuzzer] pate	ch to libkso.	so!0x2773CB	В		
		Videos										[trapfuzzer] pate	ch to libkso.	SO!0X2773CC	6 IC		
	Û	Trash										[trapfuzzer] pate	ch to libkso.	so!0x2773CF	D		
	_											[trapfuzzer] pate	ch to libkso.	so!0x4EF17F	0		
	Pa1	Floppy Disk										[trapfuzzer] pate	ch to libksol	ite.so!0x30	4EE0		
	+	Other Location	5									[trapfuzzer] pate	ch to libksol	ite.so!0x50	CFF7		
												[trapfuzzer] pate	ch to libksol	ite.so!0x50	001B		
												[trapfuzzer] pate	ch to libkso.	so!0x2773D2	E		
												[trapfuzzer] pate	ch to libkso.	so!0x2773D3	9		
												[trapfuzzer] pate	ch to libkso.	SO!0X2773D5 SO!0x2773D7	6		
												[trapfuzzer] pate	ch to libkso.	so!0x2773D9	2		
												[trapfuzzer] pate	ch to libkso.	so!0x2773DA	7		
												[trapfuzzer] pate [trapfuzzer] pate	ch to libkso. ch to libkso.	so!0x2773DB so!0x2773DC	B		
	E H											[trapfuzzer] pate	ch to libkso.	so!0x2773DE			
	wps-c	office-										[trapfuzzer] pate	ch to libkso.	so!0x2773E0	E		
	et.de	sktop										[trapfuzzer] pate	ch to libkso.	so!0x2773E1	8		
												[trapfuzzer] pate	ch to libkso.	so!0x2773E1	D		
												[trapfuzzer] pate	ch to libkso.	SO!0X2773E2	C		
												[trapfuzzer] pate	ch to libkso.	so!0x2773E3	ic		
												[trapfuzzer] pate	ch to libkso.	so!0x2773E4	0		
												[trapfuzzer] pate	ch to libkso.	50!0x2773E4 50!0x2773E4	6 R		
												[trapfuzzer] pate	ch to <u>l</u> ibkso.	so!0x2773E5	A		
												[trapfuzzer] pate	ch to				
•••																	
																0.11	
容		0 #	t 🥫	9		- 🌮 -		Q	G)					🔨 🚽 🖼 🌾 ป×	中 <u>9.11</u>	2
							_	-								202170/4	



When to kill target process

命 主页	× 🖬 pwn_iot ×	win10-wps-fuzzing	× kali-linux-20	21.1-vmware-amd64 $ imes$	🕼 kali-fuzzer 🛛 🛛	i ibuntu1804 ×
Activiti	es 🛛 🖻 WPS Spreadsh	eets 🔻				Tue 18:12 • 📇 🔩 🖒 👻
	WPS Spreadsheets	input.xls				guest ··· 🔊 – 🗗 >
0	≡ Menu ~ 🕞 🖨 🗄	₽ 80 9 0 ₹	Home Insert F	Page Layout Formulas	Data Review	View Special Features 🖓 Unsynchronized & Share 🕅 🔅 : ^
-			A ⁺ A ⁻		T	
	G & Cut	Arial				
_	Paste Copy Point	ter			Center * Te	nap hac425@ubuntu: ~/trapfuzzer-gdb/bulld 🔤 🖲 🕲 🚺
		@ fr				File Edit View Search Terminal Tabs Help
		×1 80	6			hac425@ubuntu: ~/trapf × hac425@ubuntu: ~/wps/ × hac425@ubuntu: ~/trapf × 🗷 💌
	Who Doe	s What W	here Tool	D	E IDay (Marri	[trapfuzzer] patch to libkso.so!0x23E352F
<u> </u>	2	s what, w			[Day / Mont	[trapfuzzer] patch to libkso.so!0x22E354
	3 4				UNH	[trapfuzzer] patch to libkso.so!0x23E3547 ICR Off[trapfuzzer] patch to libkso.so!0x23E35E9
S	5	1	1 3	N	1 4	Locati [trapfuzzer] patch to libkso.so!0x23E3660
	6 Sector Camp Management	Gov't Partner	Gov't Partner	Gov't Partner	Gov't Partner	Loca [trapfuzzer] patch to libkso.so!0x23E366A
	7				2000 A. 2000 A	[trapfuzzer] patch to libkso.so!0x23E36BD
	Core Relief Items	Gov't Partner, UNHCR	UNHCR	Gov't Partner, UNHCR	Gov't Partner, Shelter Aid,	Gwil[trapfuzzer] patch to libkso.so!0x23E36C2
	8				ABC AR, UNHCK	[trapfuzzer] patch to libkso.so!0x23E36CC
	9 Environment	Gov't Partner, BDC	Gov't Partner, BDC	Green Aid, Domestic Aid	Shelter Aid, Green Aid	Green [trapfuzzer] patch to libkso.so!0x23DEBC0
	Food Security					[trapfuzzer] patch to libkso.so!0x23DEC0A
	10					[trapfuzzer] patch to libkso.so!0x23DEC24
	Health	Gov't Partner, Gov't Partner,	Gov't Partner, Gov't Partner,	Gov't Partner, Gov't Partner,	Gov't Partner, Medi-Aid, Me	al-Gwil[trapfuzzer] patch to libkso.so!0x23DEDBC
	11	Gov't Partner, Gov't Partner, Health Help, Health Help, Peac	Gov't Partner, Gov't Partner, e Peace Team, IOM, IOM.	Peace Team, Health Aid, Health Aid, Health Aid, Health	Aid, Medi-Aid, Medi-Aid, Medi-Aid, Medi-Aid, UNFP/	Add B[trapfuzzer] patch to libkso.so!0x23DEDCC A Health[trapfuzzer] patch to libkso.so!0x23DEDD4
	Livelihoods	BDC, UNHCR	UNHCR	UNHCR, Domestic Aid	UNHCR	[trapfuzzer] patch to libkso.so!0x23DEDDE
	Logistics	Lors New JOM UPD UNHC	R Loss Now, UPD, UNHCR	Lors New JOM, UPD	Loss New JOM, UPD	[trapfuzzer] patch to libkso.so!0x23DEF21
	13	Legenda, toni, et bi er tre	in any room of the ordinant	UNHCR	UNHCR	UNHC[trapfuzzer] patch to libkso.so!0x23E4365
	Nutrition	Peace Team, Peace Team,	Health Aid, Child Aid, Child	Peace Team, Health Aid	FamineRelief, Medi-Aid	[trapfuzzer] patch to libkso.so!0x23E3DBE Faming[trapfuzzer] patch to libkso.so!0x23E42C2
	14	Health Aid	Aid			<pre>Child [trapfuzzer] patch to libkso.so!0x23E42D9</pre>
	Protection	Gov't Partner, Peace Team. Child Aid, UNHCR	Gov't Partner, Peace Team, BDC, Child Aid, UNHCR	Gov't Partner, Peace Team, UNHCR	Gov't Partner, Emerg Corps, Child Aid, UNHCR	Gwyf[trapfuzzer] patch to libkso.so!0x23E42DD Child/[trapfuzzer] patch to libkso.so!0x23E42EA
	Shulton	UNRED UNRED	DOA Shahar Ald UNUOD	PCA Loss New Aid or Time	Photos Aid Aid on Wime	[trapfuzzer] patch to libetmain.so!0x3E436AA
	16	UNICK, UNICK	PGA, Siener Ald, UNHCK	UNHCR	UNHCR	[trapfuzzer] patch to libkso.so!0x2753146
	Water, Sanitation &	Gov't Partner, Emerg Corps	Gov't Partner, Emerg Corps,	Gov't Partner, Peace Team,	H2O Help, H2O Help	Hen [trapfuzzer] patch to libetmain.so!0x227AB21
	17 Hygiene		UNHCR	Emerg Corps		UN A[trapfuzzer] patch to libetmain.so!0x227AB30 [IND_DDL]
	18 19 Note: Customits the sh	ove example template by a	ector and sub-sector inform	ation at northe meeting on	antion	
•••	IS Note: Customize the ab	w-to Summary	Activities Pivot Chec	k +	скиюл	
•••	nor					
ł	0		🤊 🖪 🚸	<u> </u>	5	$\land \Downarrow \cong (e, q \times q)_{2021/8/4} = (e_2)_{2021/8/4}$



Speed up Instrument

For large programs, because the number of breakpoints is very large, it will take a lot of time for each execution.

Program	desc	basic block count
WPP For Windows	Read and parse PPT file	275 1604
WPP For Linux	Read and parse PPT file	401 2478
Ichitaro 2021 Platinum	Read and parse doc、xls and so on	167 3576



Speed up Instrument - accelerated mode

In the accelerated mode, the fuzzer first obtains the executed basic blocks from the trace module, then patch the files related to these basic blocks and remove the breakpoint instructions at the corresponding positions in the files.





Lets Fuzz WPS Again!





Initial Results

Mutator	Crash Count	Time
RadamsaMutator	0	3 * 24h
TinyMutator	10+	24h

Reason

- RadamsaMutator don't consider the ratio of data mutation, it may seriously destroy the file structure, causing the testcase to be discarded very early, so that deep vulnerabilities cannot be found.
- TinyMutator can only **mutates a small percentage of the data** in the sample file, so the use cases can enter a deeper code path and the fuzzing is more efficient.



Implementation

(Version 0.3-trapfuzzer-gdb-tracer)





Why we need trapfuzzer-gdb-tracer

GdbPythonPluginTracer have some limits, such as:

- 1. It is inconvenient to debug, we need to use a python script to continuously send continue commands to the stdin of gdb.
- 2. We need to restart gdb for each test, the python runtime, and the communication overhead of fuzzer will cause additional performance overhead.

trapfuzzer-gdb-tracer has following advantages:

- 1. Written in C++, faster.
- 2. We can compile static-link GDB to reduce the requirements of environment.
- 3. Easy to debug.



GDB Internals

On the Linux platform, gdb use ptrace to debug process.

When gdb starts the debugged program, it will call start_event_loop to start the event loop and wait for events from the target process, such as hitting a breakpoint, creating a child process, and so on.

```
start_event_loop () at .././gdb/event-loop.c:370
captured_command_loop () at .././gdb/main.c:359
captured_main at ../../gdb/main.c:1202
gdb_main at ../../gdb/main.c:1217
main at ../../gdb/gdb.c:32
```



GDB Internals





GDB Internals

handle_signal_stop handles the situation where the debugged process stops due to receiving a signal,

1	static void
2	<pre>handle_signal_stop(struct execution_control_state *ecs)</pre>
3	{
4	<pre>frame = get_current_frame();</pre>
5	<pre>gdbarch = get_frame_arch(frame);</pre>
6	
7	<pre>/* Pull the single step breakpoints out of the target. */</pre>
8	<pre>if (ecs->event_thread->suspend.stop_signal == GDB_SIGNAL_TRAP)</pre>
9	{
10	// handle sigtrap signal
11	



Modify GDB

Modify the handle_signal_stop function to let gdb automatically handle the breakpoint events of the process:

First it gets the value of the pc register, and then finds the module where the pc is located
 Then according to pc and basic-block-info-file, get the original instruction of the position
 Finally, replace the breakpoint instruction with the original instruction and let the process continue execution



Modify GDB

We can use add_com to add custom gdb commands

c = add_com ("load-trapfuzzer-info", class_run, load_trapfuzzer_info, _("Load trapfuzzer config.\n"
RUN_ARGS_HELP));
set_cmd_completer (c, filename_completer);

Use execute_command_to_string to execute gdb command

```
std::string get_context_string()
```

```
std::string cmd_res = execute_command_to_string("i r", 0, false);
cmd_res += execute_command_to_string("x/4i $pc", 0, false);
cmd_res += execute_command_to_string("bt 4", 0, false);
return cmd_res;
```



Code for SIGTRAP

```
handle sigtrap event.
if (ecs->event_thread->suspend.stop_signal == GDB_SIGNAL_TRAP)
      pc = regcache_read_pc (regcache);
      // find module of pc
      COV MOD INFO* cmi = get cov mod info by pc(pc);
      // get module offset
      unsigned int voff = pc - cmi->image_base;
      // remove breakpoint
      BB_INFO* info = cmi->bb_info_map[voff];
      target write memory(pc, info->instr, info->instr size);
      if(g_debug)
          fprintf_unfiltered (gdb_stdlog, "[trapfuzzer] patch to %s!0x%X\n", cmi->module_name, voff);
      // log executed basic block
      cmi->bb_trace.push_back(voff);
      // let process continue.
      keep_going (ecs);
      return;
```



Architecture





Implementation

(Version 0.4-Windows Support)





Windows Support #1 - winappdbg-tracer

Based on winappdbg

- 1. based on python.
- 2. winappdbg basically meets the demand, but there are still some shortcomings: speed, incomplete access to the call stack.





Windows Support #2 - DbgEngTracer

The DbgEng API is a series of APIs for developing debuggers provided by Microsoft. Users only need to register the corresponding callback function to implement a debugger.

Advantage:

1. Fast execution speed.

2.DbgEng API can get the complete call stack of the program.





Windows Support #2 - DbgEngTracer

- 1. WriteVirtual/ReadVirtual: Read/Write process memory.
- 2. GetStackTrace: Get backtrace of process.





Example





Triage

crash deduplication scheme:

- 1.first get the call stack of crash
- 2.then splice the lower 12 bits of each call stack as the hash of crash
- 3.de-duplicate crash according to hash





Dialog Box

automatically handle dialog box with autoit

1	\sim	def	POPUpKillerThread(self):
2	\sim		while True:
3			<pre>time.sleep(0.1)</pre>
4			
5			<pre>ppt_text = autoit.win_get_text('Microsoft PowerPoint')</pre>
6			
7	$\mathbf{\vee}$		if u"出现严重错误" in ppt_text:
8			<pre>autoit.control_click("[Class:#32770]", "Button1")</pre>
9			
10	\sim		if u"无法编辑此" in ppt_text:
11			<pre>autoit.control_click("[Class:#32770]", "Button1")</pre>
12			
13	\sim		if u"密码" in ppt_text:
14			<pre>autoit.control_click("[Class:#32770]", "Button1")</pre>



Dialog Box

automatically handle dialog box with huorong





Other Features

Users can check the status of Fuzz through the management port during the fuzzing process

status
stage: fuzz
total dos: 59
total crash: 126
total new path: 878
speed: 18.0/min
trace mode: gdb-run
stage exec count:654
total bb count: 0
total exec time: 00 days 00h 39m 55s
coverage module: libkso.so,libwpsmain.so,libwpscloudsvrimp.so,libwpshtmlrw.so,libwpsinkdraw.so,libwpsmain.so,libwpstablestyle.so,libwpsuofrw.so,libwpswordtool.so,libwpsxmlrw.so,wp
config mutator: Vanapagan-mutator
current mutator: VanapaganMutator
last crash: 2021-06-02 20:14:13
last new path: 2021-06-02 19:50:01
current time: 2021-06-02 20:14:18
avg run time: 3.7s, timeout ratio: 5
output: //home/hac425/ndisk/doc-output
current seed file: /home/hac425/ndisk/doc-output/trapfuzz-testcase-476.bin
status (s)
seed info (si)
set seed exec count (sc)
enable debug (eb)
disable debug (db)stop
testcase (t)
crash (c)
timeout ratio (tr)
import case dir
auit (ā)

nc 127.0.0.1 8821





Start Fuzzing





Three elements of fuzzing

execution speed, seed quality/quantity, mutation algorithm

Ways to obtain samples

1. Obtained from some online sites that provide sample sets

- 2. Crawl a large number of sample files through the grammar of the search engine
- 3. Some open source projects will bring some testcases to test the program
- 4. Testcases generated when using white box Fuzz tools such as AFL to test similar software
- 5. The bug submission page of the target program or similar program
- 6. Generated with format conversion tool



Preparing the Environment

Ways for faster execution

use ramdisk, but pay attention to scheduled backups!

Ramdisk in Linux

mount -t tmpfs -o size=4G tmpfs /tmp/ramdisk/

Ramdisk in Windows

ImDisk Toolkit

٢	ImDisk ×					
Basic	Advanced Data	_				
RamDisk Configuration Tool						
	Size: G4 OKB					
Allocate Memory Dynamically						
	Drive Letter: R: V Unmount					
File System: NTFS 🗸						
☑ Launch at Windows Startup						
Create TEMP Folder						
Set TEMP Variables						
Ready						
	OK Exit					



Equipment & Results

laptop with the following configuration:

i5-8700 (4 Cores 8 threads) / 16G DDR3 RAM / 1T SSD

Software	Platform	Time	Initial Seeds	Bug Found
WPS OFFICE	Linux & Windows	≈ 8 month	Crawl	200+
中望CAD	Linux	≈ 1 day	AFL (fuzz libredwg)	4+
WPS Photo	Windows	≈ 1 day	AFL (fuzz libpng)	15
Foxit PDF	Windows	≈ 7 day	Crawl	a few crashes
Honeyview	Windows	≈ 1 day	AFL (fuzz libpng)	a few crashes
Ichitaro 2021	Windows	≈ 7 day	trapfuzzer (fuzz WPS)	100+ unique crash.



Compare with existing tools

Tool	Scenarios	Advantage	Disadvantage
AFL	Library, small program	Coverage, Speed	Large software requires wrapper and high difficulty to user
Peach	protocol fuzzing, file fuzzing	easy to use after the model is written	model development is difficult
trapfuzzer	file fuzzing now	easy to use, support large software, support coverage!	Compared with AFL, there is less feedback coverage and speed is slow.





Conclusion

- 1.Seed is important and mutate ratio is also important
- 2.GDB and DbgEng API is very nice.
- 3. Fuzzing with coverage is much better than none



Future Plans

1.More architecture support (arm, mips)

- 2.Optimize distributed Fuzz scheduling
- 3.Automatic detect data mutation ratio and mutation range.

4. More precise crash deduplication mechanism

5.....



Q&A Thank you!

Talk Title | Speaker Name here. Can change this on footer menu



References

- 1. https://loda.hala01.com/2017/06/gdb.html
- 2. <u>https://www.embecosm.com/appnotes/ean3/embecosm-howto-gdb-porting-ean3-issue-</u> 2.html